Journal of Parallel and Distributed Computing 139 (2020) 99–109

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers

Sun-Yuan Hsieh [a,*], Cheng-Sheng Liu [a], Rajkumar Buyya [b], Albert Y. Zomaya [c]

[a] Department of Computer Science and Information Engineering, National Cheng Kung University, No.1 University Road, Tainan 701, Taiwan
[b] Grid Computing and Distributed Systems (GRIDS) Laboratory, Department of Computer Science and Software Engineering, The University of Melbourne, Australia
[c] Centre for Distributed and High Performance Computing, School of Information Technologies, The University of Sydney, Sydney NSW 2600, Australia

ARTICLE INFO

Article history:
Received 10 March 2019
Received in revised form 3 September 2019
Accepted 21 December 2019
Available online 22 January 2020

Keywords:
Cloud computing
Cloud data centers
Utilization prediction model
Dynamic virtual machine (VM) consolidation

ABSTRACT

In the age of the information explosion, the energy demand for cloud data centers has increased markedly; hence, reducing the energy consumption of cloud data centers is essential. Dynamic virtual machine VM consolidation, as one of the effective methods for reducing energy energy consumption is extensively employed in large cloud data centers. It achieves the energy reductions by concentrating the workload of active hosts and switching idle hosts into low-power state; moreover, it improves the resource utilization of cloud data centers. However, the quality of service (QoS) guarantee is fundamental for maintaining dependable services between cloud providers and their customers in the cloud environment. Therefore, reducing the power costs while preserving the QoS guarantee are considered as the two main goals of this study. To efficiently address this problem, the proposed VM consolidation approach considers the current and future utilization of resources through the host overload detection (UP-POD) and host underload detection (UP-PUD). The future utilization of resources is accurately predicted using a Gray-Markov-based model. In the experiment, the proposed approach is applied for real-world workload traces in CloudSim and were compared with the existing benchmark algorithms. Simulation results show that the proposed approaches significantly reduce the number of VM migrations and energy consumption while maintaining the QoS guarantee.

1. Introduction

Cloud computing, also called Internet-based computing, provides on-demand computing resources and data to computers and any web-connected devices. By means of various technologies and concepts (e.g., hardware virtualization and data centers), cloud computing can achieve economies of scale and is grouped into three cloud service models and four cloud deployment models [24], each of which addresses a different type of business information technology (IT) requirement. In this paper, Infrastructure as a Service (IaaS) is investigated [25,29]. Various public cloud providers such as Amazon, Yahoo, and Microsoft construct enormous cloud data centers worldwide to offer cloud computing services to their customers [1,2]. The ever-growing infrastructure demands of cloud computing have resulted in remarkable growth in energy consumption at cloud data centers [6]. Excessive energy consumption not only leads to substantial operation expenses, but also produces considerable carbon emissions. Thus, the cost savings that are associated with energy conservation and effective energy-aware resource management strategies for cloud data centers have become essential, as shown in Fig. 1. Additionally, to satisfy customers' expectations concerning performance, cloud service providers and their customers must attain the required quality of service (QoS) levels. The QoS requirements are defined by service level agreements (SLAs), which are contracts that enumerate in measurable terms what services the cloud service provider must furnish, such as system throughput, response time, and down-time ratio. Consequently, reducing the power consumption in cloud data centers while preserving QoS requirements is this study's principle objective.

Generally, the average CPU utilization of physical machines (PMs) is only 15%–20% in their common state. Additionally, idle PMs comprise the majority of PMs and continuously consume 70% of their peak energy consumption [14]. Evidently, one of the principal factors in energy waste is that too many idle PMs exist. Thus, ensuring that the lowest possible number of PMs are active is an efficient approach to reducing energy expenses in cloud data centers. Recently, several studies and developments have been proposed for decreasing the energy expenses of cloud data

* Corresponding author.
E-mail addresses: hsiehsy@mail.ncku.edu.tw (S.-Y. Hsieh), eric81115@gmail.com (C.-S. Liu), raj@csse.unimelb.edu.au (R. Buyya), albert.zomaya@sydney.edu.au (A.Y. Zomaya).

https://doi.org/10.1016/j.jpdc.2019.12.014
0743-7315/

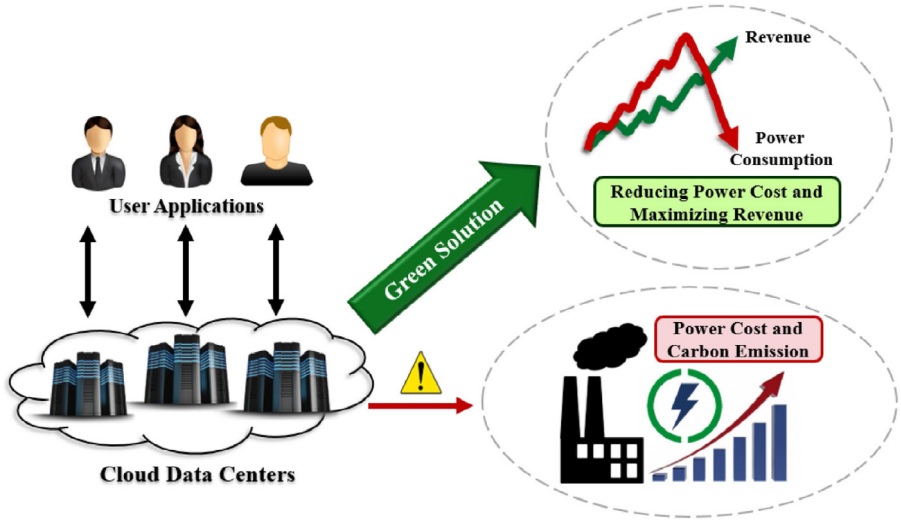


Fig. 1. Green cloud computing.

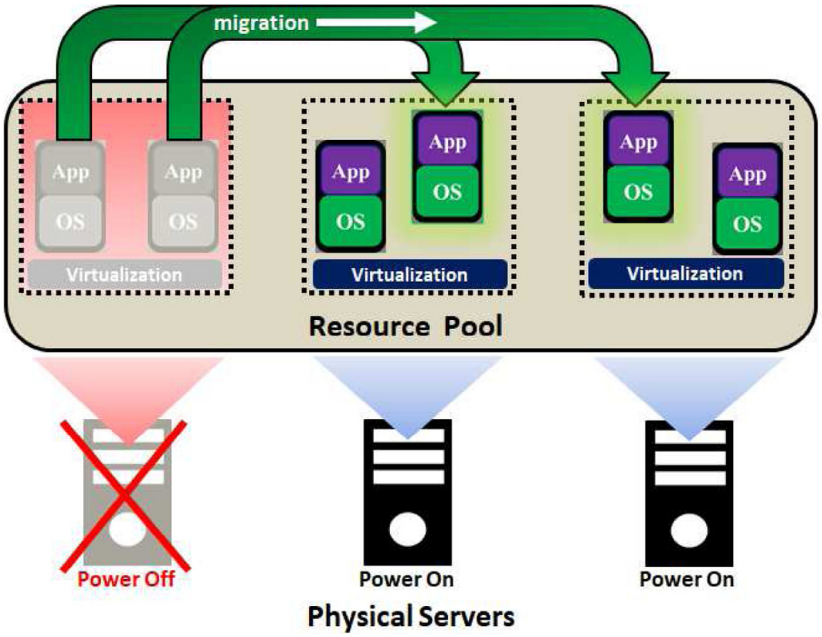


Fig. 2. The VM consolidation schematic diagram.

centers. Dynamic virtual machine (VM) consolidation is an effective approach for decreasing the energy expenses and resource utilization [6,9]. By leveraging hardware virtualization technology [3], several VMs are hosted on the same physical server, on which each VM can run one or more applications. Moreover, the hardware virtualization enables individual tasks to fewer servers to optimize the resources efficiency. By using live VM migration techniques, VMs can be consolidated and packed on fewer PMs, thereby reducing the energy consumption [8]. The VM consolidation schematic diagram is presented in Fig. 2. However, the workload of diverse application types is dynamically variable. To most effectively utilize the resources, the VM consolidation approach must be employed in online.

VM consolidation is commonly split into the following steps [5]: (1) detecting overloaded hosts; (2) detecting under-loaded hosts; (3) selecting VMs; and (4) placing VMs. Our study focuses on the first and second steps of the VM consolidation problem.

In detail, when a host is identified as overloaded, some of the VMs on said host should be properly selected for migration to other suitable hosts. If no host in an active state with adequate resources to run the VM, an inactive host is initiated so that the selected VMs can be allocated to that machine. In addition, when a host is identified as under-utilized, all VMs from that host are chosen for migration if they can be consolidated into other suitable hosts without exceeding the full load. Furthermore, to save energy, idle hosts are subsequently switched into a low-power state. However, switching the power state of a host from idle to low-power state and vice versa wastes additional energy [15,30]. Hence, to save power, switching hosts' states is necessary, but restricting their frequency is even more vital.

In this paper, our approach is used to predict short-term future resource utilization on the basis of historical data on the sample hosts. Current and predicted utilization metrics are considered a reliable characterization of overloaded and under-loaded hosts. Because CPU utilization has the greatest impact on energy expense [4,5], our approach focuses on CPU utilization in relation to CPU resources. Our main contributions are described as follows:

- An effective utilization prediction approach based on the Gray–Markov forecasting model to forecast short-term future CPU utilization according to the accumulated data on the considered hosts is proposed. In addition, combining data on present and near future CPU utilization is a dependable characterization of overloaded and under-loaded hosts. Thus, cloud providers can increase energy efficiency and the SLA's performance guarantee.
- Effective dynamic VM consolidation with a utilization prediction algorithm for energy-efficient cloud data centers is proposed, namely, utilization prediction-based potential overload detection (UP-POD) and utilization prediction-based potential underload detection (UP-PUD).
- A power-saving value based on power consumption and number of migrations for detecting under-loaded hosts is proposed. This value can be used to more reliably select under-loaded hosts.
- Through several simulations based on real-world workloads, the proposed approach reduces the energy consumption while restricting the number of migrations. Therefore, it increases the performance of cloud data centers with an improved SLA performance guarantee.

The remainder of this paper is organized as follows: Section 2 summaries related studies on the dynamic VM consolidation problem in cloud data centers. The problem statement is provided in Section 3. The system architecture, Gray–Markov-based prediction model, and the VM consolidation algorithms comprising host overload detection and host underload detection are presented in Section 4. The implementation instructions for our approach are presented in Section 5. Finally, the experimental results and conclusion are illustrated in Sections 6 and 7, respectively.

2. Related works

In this section, studies on the problem of dynamic VM consolidation problem in cloud data centers are discussed. Numerous studies have sought to address this problem.

The problem of decision-making when a host is overloaded or under-loaded has been investigated in the literature [6,21,32,33]. The critical concern is the decision of whether a host is considered overloaded or under-loaded depending on the variations in VM workload over time and types of user applications. When determining whether a host is overloaded or under-loaded, several VM consolidation approaches have considered only current resource utilization [4,21]. Such approaches only may lead to needless migrations, thereby aggravating the overhead such as the energy costs for VM migration, performance degradation attributable to migration, and extra traffic [12,15,30]. Therefore, the threshold should be robustly decided to restrict the frequency of VM migration.

Several studies have addressed dynamic VM consolidation by applying migration techniques to optimize power consumption [5,23,33,34]. In the primary method, thresholds set statically were applied to decide whether a host is overloaded or under-utilized. These approaches attempt to maintain the current utilization of a host between the hot and cold thresholds. Zhu et al. [34] proposed a method for determining the static threshold of CPU utilization to estimate when a host is considered overloaded. If CPU utilization exceeds 85%, the host is considered overloaded. However, setting static thresholds and using the current resource usage are not effective measurement approaches for cloud data centers with dynamic workloads, where the utilization of VMs running on a physical server continuously changes. Beloglazov et al. [5] proposed a set of adaptive upper thresholds (i.e., median absolute deviation (MAD), interquartile range (IQR), and local regression (LR)) that can be obtained through statistical analysis of the historical data. Although the considered thresholds are not static values, these approaches use only current resource utilization as the principal criterion to make decisions regarding VM migrations. Thus, they cannot make reliable decisions when the host load needs to be reduced, and this causes energy waste and unnecessary migrations. In the literature [11,19,33], the problem of forecasting future resource usage by utilizing historical data in cloud data centers has been studied. Farahnakian et al. [11] proposed a method employing linear regression to predict CPU usage for VM consolidation. Jheng et al. [19] applied the Gray prediction model to predict host CPU and RAM resource utilization. The Gray forecasting model does not require substantial training data and is based on simple mathematical derivations. The experimental results indicated that this model cannot guarantee dependable prediction results for workloads with frequent fluctuations. Moreover, Markov prediction is effective for predicting statistical data with frequent fluctuations. To improve the prediction accuracy of the Gray forecasting model, adding Markov prediction can enhance systems with severe randomly variable time series. Therefore, in cases of time series with frequent fluctuations, Gray–Markov prediction model performs more accurately than does the Gray prediction model [17]; hence the Gray–Markov model is considerably more suitable for the dynamic workload of cloud data centers.

The Gray–Markov prediction model has been used in many fields such as electricity demand [31], fire accidents [22], and traffic volume [29]. No studies have implemented the Gray–Markov prediction model for analyzing resource demand in cloud data centers. Thus,we did so in this study to optimize the Gray forecasting model.

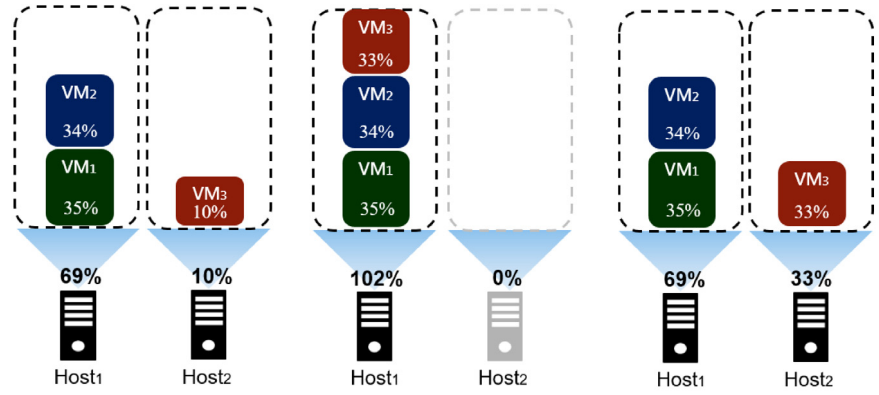
3. Problem statement

The problem is presented in Fig. 3. Suppose that a large cloud data center supplies computing resources in the form of VM instances. In this paper, the VM consolidation problem concerns the decision regarding a host is overloaded or under-loaded. Once the overloaded hosts have been identified, all VMs with a potential increase in CPU utilization are migrated from these hosts to maintain QoS; once the under-loaded hosts have been identified, all VMs from this host are migrated from these hosts to reduce energy consumption.

An efficacious VM consolidation approach optimizes VM placement for the maximum expected benefit to reduce the number of hosts in an active state. The benefit originates from two main factors: the number of VM migrations and the rate of SLA violations. More crucially, by assigning VMs to hosts on the basis of their near-future resource utilization, these benefits can be realized in advance. To describe the problem of the conventional VM consolidation approach and the benefits of the prediction-based approach, example is represented in Fig. 3.

Suppose that there are two hosts and three VMs.

1. At time t, the CPU utilization of $Host_1$ and $Host_2$ is 0.35 and 0.60, respectively. Because $Host_1$ has sufficient resources to run VM_3, a normal VM consolidation migrates VM_3 to $Host_1$ to minimize the number of hosts in an active state and switches $Host_2$ into a low-power state.
2. At time $t + 1$, the CPU utilization requested by VM_3 increases from 0.60 to 0.75. Because $Host_1$ has inefficient free capacity to contain VM_3, $Host_1$ is overloaded and some SLA violations occur.
3. At time $t+2$, VM_3 migrates to $Host_2$ to avoid additional SLA violations. Hence, if a VM consolidation approach predicts the resource requirements of a VM before migration, the unnecessary migrations can be evaded and the rate of SLA violations can be reduced.



(a) At time t: VM_3 migrates from $Host_2$ to $Host_1$. (b) At time $t+1$: VM_3 migrates from $Host_1$ to $Host_2$. (c) At time $t+2$: no migration.

Fig. 3. Problem statements for Example.

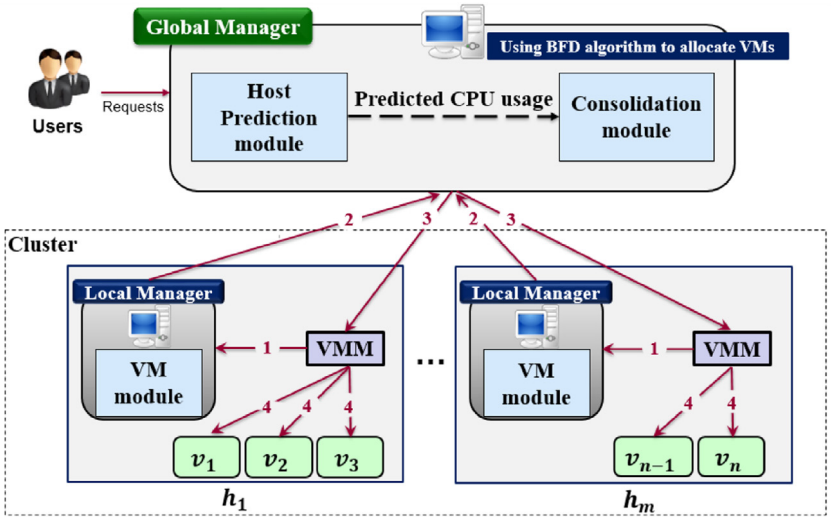


Fig. 4. System architecture.

4. Proposed utilization-prediction-aware VM consolidation

In this section, our proposed utilization-prediction-aware VM consolidation approach for cloud data centers is divided into several parts. In Sections 4.1 and 4.2, the system architecture of the cloud data center and the Gray–Markov-based prediction model is presented. In addition and most crucially, the resource utilization prediction algorithms (UP-POD and UP-PUD) and power saving value based on the predicted CPU utilization are presented in Section 4.3.

4.1. System architecture

Our system architecture is presented in Fig. 4.

Our implementation consists of m heterogeneous hosts (i.e., $H = \langle h_1, h_2 \ldots, h_m \rangle$) in a cloud data center. Each host is characterized by different resource types such as CPU, memory size, network bandwidth, and storage capacity. Additionally, CPU is usually measured in million instructions per second (MIPS). At any given time, many simultaneous users use the services of a cloud data center. The provisioning of n VMs (i.e., $V = \langle v_1, v_2 \ldots, v_n \rangle$) is requested by users. The VMs are initially allocated to hosts applying the best fit decreasing (BFD) algorithm, which is one of the most widely used heuristic algorithms for solving the bin-packing problem. Because of the BFD algorithm, all unutilized space in the destination hosts is minimized. The algorithm selects a host for which the amount of available resources is closest to the amount of resources requested by the VM. This explains why the BFD algorithm effectively performs the initial allocation of VMs. However, the requested utilizations of running hosts and VMs change over time due to dynamic workloads with frequent fluctuation: hence the initial allocation approach must be enhanced with a VM consolidation algorithm that can be implemented periodically to optimize the performance of cloud data centers. Our proposed approach is conducted every 5 min in a cloud data center to reduce energy expenses and the number of hosts in active state.

The system architecture comprises two types of agents: (1) a global manager (GM) deployed in a master node, and (2) entirely distributed local managers (LMs) in all hosts. The two agents execute the following steps at each iteration:

1. The current resource utilization of all VMs in a host is monitored by each LM periodically. Each LM predicts the future CPU utilization of a host on the basis of historical data in a log file by applying the Gray–Markov prediction model precisely.
2. The information and status from the LMs are gatherd by the GM to comprehend the overall situation of hosts

(i.e, current and future CPU resource utilization and numbers of VMs running on each host).
3. The GM sends migration commands to the virtual machine monitor (VMM) to perform the UP-POD and UP-PUD algorithms of our proposed approach. The commands indicate which VMs should be migrated to which destination hosts on the basis of the consolidation algorithms.
4. After receiving the commands from the GM, the VMMs migrate the VMs.

4.2. Prediction method

4.2.1. GM(1, 1) grey prediction model

Among the family of gray forecasting models, the most frequently used is the GM(1, 1) model [10]. Conventional forecasting techniques usually handle original historical data series directly and attempt to approximate their evolutionary behavior. However, through a preliminary transformation, the initial GM(1, 1) model starts by converting the original data series into a monotonically increasing data series; this is called an accumulated generating operation (AGO). By applying the AGO technique, the noise of the original data series is efficiently reduced, and the generated new data series exhibits exponential behavior approximately. Because the solution of first-order differential equations also takes the exponential form, the first-order gray differential equations are constructed to model the data series from the AGO and forecast the future behavior of the system. Generally, the procedure of a Gray model is derived as follows:

Step 1: Assume the host's historical data on CPU utilization with n samples (time point) as

$$X^{(0)} = \{x^{(0)}(1), x^{(0)}(2), \ldots, x^{(0)}(n)\}, \tag{1}$$

Step 2: Construct the AGO. Let $X^{(1)}$ be the transformation sequence of $X^{(0)}$:

$$X^{(1)} = \{x^{(1)}(1), x^{(1)}(2), \ldots, x^{(1)}(n)\}, \tag{2}$$

where

$$X^{(1)}(k) = \sum_{i=1}^{k} x^{(0)}(i), \quad k = 1, 2, 3, \ldots, n. \tag{3}$$

Consequently, the model of the first-order differential equation GM(1, 1) is

$$\frac{dx^{(1)}}{dt} + ax^{(1)} = b, \tag{4}$$

where t is independent variable, a is the development coefficient, and b is the gray control variable.

Step 3: Perform the conversion treatment on the former and latter terms. Consider the following:

$$\frac{dx^{(1)}}{dt} \longrightarrow x^{(1)}(k+1) - x^{(1)}(k). \tag{5}$$

Through an inverse AGO (IAGO), it can be derived that

$$x^{(1)}(k+1) - x^{(1)}(k) = x^{(0)}(k+1). \tag{6}$$

Its definition in $x_1^{(1)}(t)$ is

$$x^{(1)}(k) \longrightarrow 0.5x^{(1)}(k) + 0.5x^{(1)}(k-1) = z^{(1)}(k). \tag{7}$$

After collation, it is known that

$$\frac{dx^{(1)}}{dt} + ax(1) = b \longrightarrow x^{(0)}(k) + az^{(1)}(k) = b. \tag{8}$$

Step 4: Determine a, b by using the least squares method. Consider the following:

$$\begin{bmatrix} a \\ b \end{bmatrix} = (B^T B)^{-1} B^T Y_n \tag{9}$$

where the accumulated matrix B and constant term Y_n are

$$Y = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \vdots & 1 \\ -z^{(1)}(n) & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \vdots \\ x^{(0)}(n) \end{bmatrix}. \tag{10}$$

Step 5: Construct the gray prediction model. The equation in GM(1, 1) is

$$x^{(0)}(k) + az^{(1)}(k) = b \tag{11}$$

where the primitive condition of $x^{(1)}$ is $x^{(0)}(1) = x^{(1)}(1)$.

The whitening equation is expressed as follows:

$$\frac{dx^1(t)}{dt} + ax^{(1)}(t) = b. \tag{12}$$

Step 6: According to Eq. (12), the solution of $x^{(1)}(t)$ at time k is

$$\widehat{x}^{(1)}(k+1) = [x^{(0)}(1) - \frac{b}{a}]e^{-ak} + \frac{b}{a}. \tag{13}$$

To obtain the predicted value of the primitive data at time (k + 1), the IAGO is used to establish the following gray model:

$$\widehat{x}^{(0)}(k+1) = [x^{(0)}(1) - \frac{b}{a}]e^{-ak}(1 - e^{a}). \tag{14}$$

and the predicted value of the primitive data at time (k + H):

$$\widehat{x}^{(0)}(k+H) = [x^{(0)}(1) - \frac{b}{a}]e^{-a(k+H-1)}(1 - e^{a}). \tag{15}$$

4.2.2. Gray–Markov method

The Gray–Markov prediction model combines the Gray model and Markov chain [20,22]. In this approach, the initial prediction is employed by applying Grey prediction model; furthermore, the Markov chain is implemented to predict the error in prediction determined using the Gray prediction model. By adding this predicted error to the predicted outcome of the Gray system, the whole prediction error decreases one step further and the prediction accuracy increases markedly. The procedure for integrating the Markov method into forecasting is presented as follows.

1. *Determining the state of error*
   In the first step, errors in the Gray prediction model for several previous predictions considered (in this paper, 24 previous prediction errors are considered). The interval of these states is equal and defined as
   $$R_i = \frac{(err_{max} - err_{min})}{NS} \tag{16}$$
   where err_{max} and err_{min} are the maximum and minimum values of the samples, respectively, and NS is the number of states.
2. *Constructing the probability transition matrix*
   A procedure by which the nth sample from state i changes to the $(n+1)$th sample at state j is the transition T_{ij}. Higher orders of transition $T_{ij}^{(m)}$ are defined when the nth sample is in state i being changed to the $(n+m)$th sample, which is in state j. The probability of the transition from state i to state j can be determined using the following equation:
   $$P_{ij}^{(m)} = \frac{M_{ij}^{(m)}}{M_i} \tag{17}$$
   where i and j may vary from 1 to the number of states. $M_{ij}^{(m)}$ and M_i denote the number of transitions from state i to state j and the number of data items in state i, respectively. The matrices constructed from various orders of transition probability are viewed as transition probability matrices.

Table 1
Notations.

H	Set of hosts
H_{active}	Set of active hosts
H_{over}	Set of overloaded hosts
H_{under}	Set of candidate for under-loaded hosts
V	Set of VMs
$L_h(t)$	Load of the host h at time t
$C_h(t)$	Total CPU capacity of the host h at time t
$U_h(t)$	CPU utilization of the host h at time t
$D_h(t)$	The length of historical data of the host h at time t
P_h	Power consumption of the host h
$\widehat{P}_h$	Predicted power consumption of the host h
S_h	Power saving value of the host h
M_h	Number of VMs running on the host h
t_u	The upper threshold value of CPU utilization
t_l	The lower threshold value of CPU utilization

Each array displays the probability of m order transition from state i to state j:

$$P^{(m)} = \begin{bmatrix} P_{11}^{(m)} & P_{12}^{(m)} & \cdots & P_{1n}^{(m)} \\ P_{21}^{(m)} & P_{22}^{(m)} & \cdots & P_{2n}^{(m)} \\ \vdots & \vdots & \vdots & \vdots \\ P_{n1}^{(m)} & P_{n2}^{(m)} & \cdots & P_{nn}^{(m)} \end{bmatrix} \tag{18}$$

3. *Determining high-probability transitions*
   Several flow-order probability transition matrices (the third-order probability matrix is considered the highest order in this paper) are applied. Moreover, the highest probability of transition from the preceding sample is acquired.
4. *Forecasting the prediction error*
   Finally, the state where the next sample is most likely to be, is applied to predict the error. Using the following equation, which calculates the average state decided in Step c, the number of predicted samples are obtained. Consequently, by adding this value to the predicted value from the Gray prediction model, the outcome of the final prediction is produced as follows:

$$\widehat{X}_{GM}^{(0)}(k+1) = X_G^{(0)}(k+1) + \frac{1}{2}(A+B) \tag{19}$$

where $\widehat{X}_{GM}^{(0)}$ denotes the predicted value using the Gray–Markov prediction model and $\widehat{X}_G^{(0)}$ denotes the predicted value by using only the Gray prediction model. In addition, A and B are end points of the interval that describes the future state of error of the Gray prediction model, which is calculated in step c.

4.3. The resource utilization prediction algorithm

The preliminary notations and definition employed in our proposed algorithms are listed in Table 1.

4.3.1. Utilization prediction-based overload detection

In every dynamic VM consolidation process, each host must be identified whether overloaded or not. As motivated by [18], the proposed utilization prediction based potential overload detection (UP-POD) is presented in Algorithm 1.

The input of Algorithm 1 is a set of active hosts H_{active}. For every host in H_{active}, which are overloaded is determined. Subsequently, the overloaded hosts are added into H_{over} as output to execute the migration decision. Algorithm 1 can be explained step by step as follows.

In line 1, the GM receives $U_h(t)$, which is defined as $L_h(t)$ divided by $C_h(t)$. In line 2, by obtaining the historical data on CPU utilization of a host h recorded in a log file, the short-term utilization of CPU (i.e., $U_h(t+1)$) can be calculated using a time-series-based forecasting model. Gray–Markov time-series model is used to predict $U_h(t+1)$ as the output $\widehat{X}_{GM}^{(0)}(k+1)$ presented in Section 4.2. Furthermore, the input of our time series data is the historical data on CPU utilization recorded at 5-min intervals in each host. In line 3, after preprocessing, the decision is made in accordance with the dynamic upper threshold method. In this method, if a host's utilization is higher than the upper threshold, it is considered overloaded. The dynamic upper threshold is set by applying the median absolute deviation (MAD) approach presented in [5], and the parameter s is 2.5 in conformity with [5]. In lines 5–9, to forecast the $U_h(t+1)$ precisely, the time-series model requires sufficient historical data for computation. If insufficient historical CPU utilization data on each host is available, the decision can only using $U_h(t)$. During the simulation, our experiments tested historical data with lengths of 12, 16, 20, 24, and 28. On the basis of the results, the proposed algorithms perform optimally with historical data with a length of 24. Thus, if the length of the historical data $d_h(t)$ is less than 24, $U_h(t)$ is considered to make the decision. The hosts would be considered overloaded and added into H_{over} if their $U_h(t)$ values are higher than t_u. By contrast, in lines 10–13, $d_h(t)$ is at least 24, the host is considered overloaded and is added into H_{over} if the present and predicted short-term values of CPU utilization are higher than t_u (i.e., $U_h(t) > t_u$ and $U_h(t+1) > t_u$). This situation demonstrates that the host is a potential candidate that executes the migration decision when it is overloaded in both the present and near future.

Consequently, Algorithm 1 considers not only the present situation but also the near future situation. Algorithm 1 can prevent unnecessary migrations to reduce the overall number of migrations and execute an appropriate migration decision; moreover, the SLA violation rate can be maintained in advance.

```
Input: H_active
Output: H_over
U_h(t)=L_h(t)/C_h(t) ; /* RequiredMIPS(t)/TotalMIPS(t)   */
predict U_h(t+1); /* using Grey-Markov prediction     */
set t_u; /* applying MAD=1-s*Mad                       */
foreach h ∈ H_active do
    if D_h(t) < 24 then
        if U_h(t) > t_u then
            return true;
        end
        else
            return false;
        end
    end
    if U_h(t) > t_u and U_h(t+1) > t_u then
        return true;
    end
    else
        return false
    end
end
```

Algorithm 1: UP-POD

4.3.2. Utilization prediction-based underload detection

After overloaded hosts have been identified, the underload detection algorithm commences. To reduce the number of hosts in an active state to reduce, thereby reducing energy consumption, determining which host is under-loaded and switching it into low-power mode is essential.

Table 2
Workloads data characteristics (CPU utilizaton).

Workloads	Date of Workloads	Number of servers	Number of VMs	Mean	St.dev.	Quartile 1	Median	Quartile3
W1	03/03/2011	800	1052	12.31%	17.09%	2%	6%	15%
W2	06/03/2011	800	898	11.44%	16.83%	2%	5%	13%
W3	09/03/2011	800	1061	10.70%	15.57%	2%	4%	13%
W4	22/03/2011	800	1516	9.26%	12.78%	2%	5%	12%
W5	25/03/2011	800	1078	10.56%	14.14%	2%	6%	14%
W6	03/04/2011	800	1463	12.39%	16.55%	2%	6%	17%
W7	09/04/2011	800	1358	11.12%	15.09%	2%	6%	15%
W8	11/04/2011	800	1233	11.56%	15.07%	2%	6%	16%
W9	12/04/2011	800	1054	11.54%	15.15%	2%	6%	16%
W10	20/04/2011	800	1033	10.43%	15.21%	2%	5%	12%

The proposed UP-PUD is presented in Algorithm 2. The input of Algorithm 1 is a set of active hosts H_{active}. For every host in H_{active}, which hosts are candidates of under-loaded hosts is determined, and these hosts are subsequently added to H_{under} as output. Algorithm 2 can be explained step by step as follows. The procedure and concept of Algorithm 2 are similar to those of Algorithm 1. The difference is that in lines 5–9, if $U_h(t)$ is less than or equal to t_l, the host is added into H_{under}. Additionally, in lines 10–13, $d_h(t)$ is at least 24, and the host is considered under-loaded and added into H_{under} if the present and predicted short-term values for CPU utilization are less than or equal to t_l (i.e., $U_h(t) > t_l$ and $U_h(t+1) > t_l$).

Input: H_{active}
Output: H_{under}
$U_h(t)$=$L_h(t)/C_h(t)$; /* RequiredMIPS(t)/TotalMIPS(t) */
predict $U_h(t+1)$; /* using Grey-Markov prediction */
set t_l; /* $t_l = 30\%$ */
foreach $h \in H_{active}$ **do**
  if $D_h(t) < 24$ **then**
    if $U_h(t) \leq t_l$ **then**
      return *true;*
    end
    else
      return *false;*
    end
  end
  if $U_h(t) \leq t_l$ **and** $U_h(t+1) \leq t_l$ **then**
    return *true ;*
  end
  else
    return *false*
  end
end

Algorithm 2: UP-PUD

After candidates of under-loaded hosts are selected and added into H_{under} by Algorithm 2, the proposed S_h value is applied to select the final under-loaded host from H_{under}.

R. Nathuji et al. [27] and X. Fu et al. [13] have explained that the host's power consumption is near proportional to its CPU utilization. Therefore, the power consumption of each host can be calculated using Eq. (20):

$$P(\mu) = 0.7 * P_{max} + 0.3 * P_{max} * \mu \tag{20}$$

where notation P_{max} denotes the host's power consumption value when it is in full load. The notation μ denotes the host's CPU utilization, which is observed variably. Therefore, the host's CPU utilization is mainly considered for employment in our VM consolidation approach.

In [16], a power-efficient value for a host in an active state. This value can be applied to select under-loaded host. On the basis of our prediction model, a power-saving value (S_h) is proposed in Eq. (21) by improving the power-efficient value. This value can be used to more precisely detect under-loaded hosts.

$$S_h = \frac{P_h + \widehat{P_h}}{M_h} \tag{21}$$

In Eq. (21), P_h represents the power consumption of the hth host in the cloud data center, $\widehat{P_h}$ represents the power consumption of the hth host by using $U_h(t+1)$ for calculation, and M_h represents the number of VMs running on the hth host.

Finally, the host with the maximal S_h value can be chosen as the under-loaded host. Evidently, because S_h has considered only the host'sôspresent power consumption, power consumption at time t+1, and the number of VM migrations into consideration, it will be more efficiently when detecting an under-loaded host.

5. Experimental setup

In this section, workload types, the simulation environment, and performance metrics are implemented to analyze the performance of our proposed approach.

5.1. Workload data

For effective comparison with [5], our simulation uses workloads of the same 10-day period. The CPU utilization of the VMs corresponds to their workloads and their statistical analysis is described in Table 2. The experiments are implemented using real-world publicly available workloads, in the form of PlanetLab data [28] provided as a portion of the CoMon project: a monitoring infrastructure for PlanetLab. The workload data comprise CPU utilization of a VM logged at 5-min intervals and were measured on 10 different days during March and April 2011. Each VM contains 288 records on CPU utilization, and the records are plugged into dynamic VM consolidation. Additionally, the data are assembled from more than 1000 VMs hosted on servers in more than 500 locations worldwide. In reality, the workload is representative of an IaaS cloud environment such as Amazon EC2, where VMs are controlled and created by individual users.

5.2. Simulation environment

To impartially compare the efficiency of the proposed time series prediction of our short-term-based VM consolidation approach, the experiment employs the CloudSim 3.0.3 toolkit [7]. Our simulation involved a data center comprising 800 heterogeneous PMs. Half of the PMs are HP ProLiant ML110 G4 servers with 1860 MIPS per core, and the other half are HP ProLiant ML110 G5 servers with 2660 MIPS per core in each workload. In detail, each PM is modeled to have two cores, 4 GB of memory and 1 GB/s of network bandwidth. Table 3 specifies the CPU MIPS rating and memory amount characteristics of the four VM instances employed in CloudSim corresponding to Amazon EC2 [3].

Table 3
VM details.

VM type	CPU (MIPS)	RAM (GB)
High-CPU medium instance	2500	0.85
Extra-large instance	2000	3.75
Small instance	1000	1.7
Micro instance	500	0.613

5.3. Performance metrics

The objectives of our approach are: (1) to reduce power consumption; (2) to reduce the SLA violation rate; (3) to reduce the number of hosts in an active state; (4) to reduce the number of migrations. Hence, the following metrics are used to assess the performance of the proposed approach:

- **SLA Violations:**
  To maintain the QoS guarantee in an IaaS between cloud service providers and users, thereby achieving an agreeable SLA, Eq. (22) can be used to judge the quality of the cloud service. The rate of SLA violations is measured using two metrics: SLA violations due to over-utilization (SLAVO) and SLA violations due to migration (SLAVM):

  $$SLAV = SLAVO \times SLAVM \tag{22}$$

  where SLAVO represents the average ratio for the period when the host experiences 100% CPU utilization, as shown in Eq. (23):

  $$SLAVO = \frac{1}{M}\sum_{i=1}^{M}\frac{T_{s_i}}{T_{a_i}} \tag{23}$$

  where M denotes the number of hosts and T_{s_i} denotes the total time host i has experienced the 100% CPU utilization that causes an SLA violation. The notation T_{a_i} denotes the time at which host i is in an active state. SLAVM represents the overall performance degradation by VMs due to migrations, as shown in Eq. (24):

  $$SLAVM = \frac{1}{N}\sum_{j=1}^{N}\frac{C_{d_j}}{C_{r_j}} \tag{24}$$

  where N denotes the number of VMs, C_{dj} denotes the performance degradation caused by migrating VM, and C_{rj} denotes the total CPU utilization requested by VM j during its lifetime.
- **Energy consumption:**
  Most studies have determined that CPU resources uses more power consumption than memory, network interface, or disk storage. Measuring energy consumption is based on real data on SPECpower benchmark results [5]. Table 4 illustrates that at different load levels, energy consumption in HP ProLiant G4 servers and HP ProLiant G5 servers changes. Notably, when under-utilized servers enter the low-power state, energy consumption decreases significantly. Therefore, reducing the number of hosts in an active state is necessary.
- **Number of VM Migrations:**
  Live VM migration incurs extra expenses and additional performance degradation such as the source host's extra CPU utilization, extra network bandwidth, applications downtime during VM migrations, and whole migration time [26]. Therefore, reducing the number of VM migrations is essential because this likely causes SLA violations.
- **Energy and SLA Violations (ESV):**
  The main objective of the proposed VM consolidation approach is to simultaneously reduce energy expenses and SLA violations. Because a trade-off occurs between energy consumption and performance, a combined metric called Energy and SLA Violations that can be used to effectively judge the trade-off is shown in Eq. (25):

  $$ESV = E \times SLAV \tag{25}$$

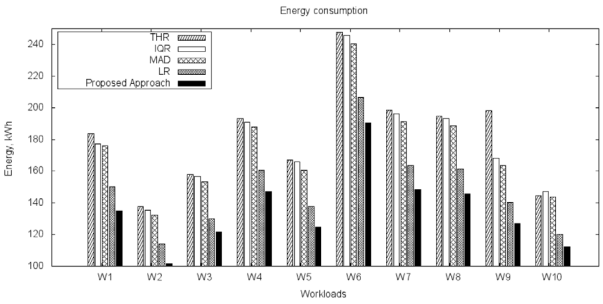


Fig. 5. Comparison of energy consumption for 10 workloads.

5.4. Comparison benchmarks

For efficient verification, the proposed approach is compared with the algorithms proposed for detecting overloaded hosts as follows [5]. These algorithms are presented in the CloudSim simulator [7].

1. Static threshold (THR): the hot threshold value is set at 90%. If the current CPU utilization of hosts exceed 90%, the hosts are considered overloaded.
2. Two adaptive thresholds: the median absolute deviation (MAD) and interquartile range (IQR). The algorithm functions identically to the THR. The detailed calculation of MAD and IQR is presented in [5].
3. A dynamic threshold called the local regression (LR) method [5]: Overloaded hosts are decided according to the calculation of local regression changes over time.

6. Experimental results

In this section, the proposed approach is compared with the four benchmark algorithms introduced in Section 5.4. The VM selection policy is the maximum correlation policy proposed in [5]. Figs. 5–10 present the comparison results of ten days workloads. Each workload is evaluated over a time span of 24 h.

Fig. 5 illustrates a comparison of performance according to the energy consumption metric. The proposed approach reduces energy consumption by an average of 25.6%, 23.7%, 22.4%, and 9.6% compared with THR, IQR, MAD, and LR, respectively. By employing UP-PUD and the power-saving value to identify under-loaded hosts, such hosts can be selected more precisely. After identifying the under-loaded hosts, all VMs in these hosts can be migrated to other suitable hosts, and the host could be switched into sleep mode. Consequently, energy can be saved by switching idle hosts to low-power states during the consolidation process. Fig. 6 presents the comparison of performance with regard to the SLAVO metric. Compared with THR, IQR, MAD, and LR, the performance does not exhibit improvements. Efforts made to most efficiently maximize the resource use of the hosts and inaccuracy attributable to the approach implemented based on prediction may explain the lack of improvement. However, the SLAV metric is the multiplication of SLAVO and SLAVM metrics, owing to our performance in SLAVM metric is pretty good, the performance in SLAVO metric somewhat has not apparent improvement that can be negligible in the proposed algorithm. This can be clearly observed by analyzing performance according to

Table 4
Power comsumption of the selected servers at different load levels (in Watts).

Server	Sleep mode	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProLiant G4	10	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant G5	10	93.7	97	101	105	110	116	121	125	129	133	135

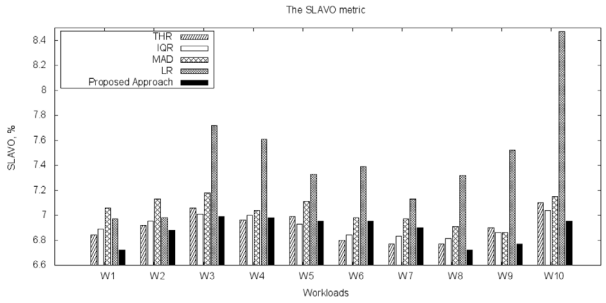


Fig. 6. Comparison of the SLAVO metric for 10 workloads.

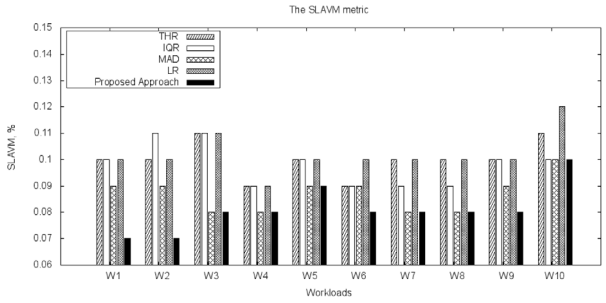


Fig. 7. Comparison of the SLAVM metric for 10 workloads.

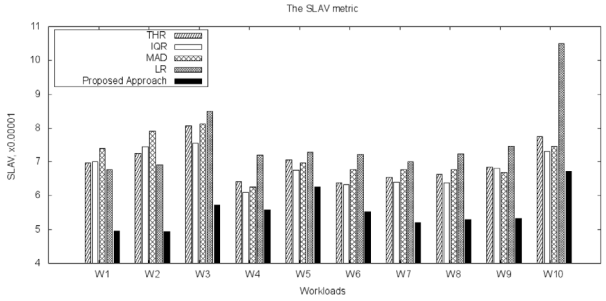


Fig. 8. Comparison of the SLAV metric for 10 workloads.

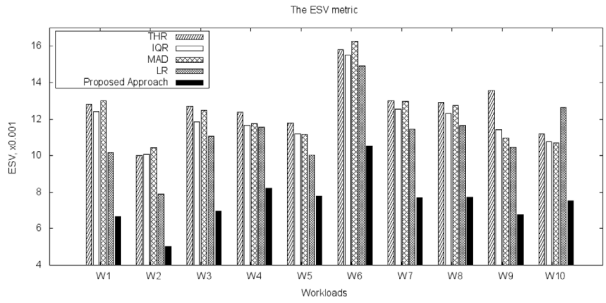


Fig. 9. Comparison of the ESV metric for 10 workloads.

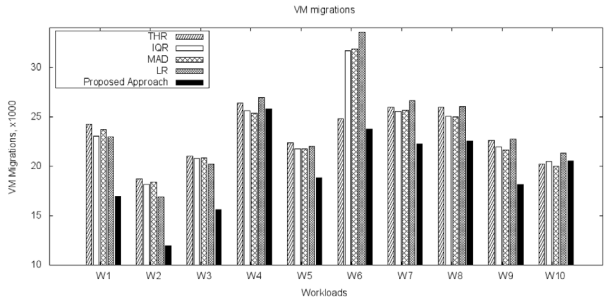


Fig. 10. Comparison of number of VM migrations for 10 workloads.

the SLAV metric. Fig. 7 presents the comparison of performances with regard to the SLAVM metric. Compared with THR, IQR, MAD, and LR, the performance exhibits a greater improvement because our approach resulted in a substantial reduction in the number of VM migrations.

Fig. 8 presents the performance comparison according to the SLAV metric. The proposed approach reduces SLA violation rate by an average of 21.6%, 18.3%, 21.7%, and 26.7% compared with THR, IQR, MAD, and LR, respectively, because our prediction-based approach performs on the SLAVM metric.

Fig. 9 presents the performance comparison according to the ESV metric. The proposed approach reduces energy consumption by an average of 42.7%, 38.1%, 39%, and 33.1% compared with THR, IQR, MAD, and LR, respectively. Our approach causes such considerable improvement because of the reductions in energy consumption and the SLAV violation rate. In truth, these notable results indicate that our approach involves a successful trade-off between power cost and QoS guarantee. Fig. 10 presents the performance comparison based on the number of migrations. Compared with THR, IQR, MAD, and LR, performance improved. Through UP-POD and UP-PUD, the number of VM migrations was considerably reduced. The prediction-based algorithm and power-saving value detect hosts that are likely to overload and underload in the near future. Thus, they prevent repeated migrations of VMs.

Overall, our proposed approach significantly outperforms real-world benchmarks applications.

7. Conclusions

In this paper, the dynamic VM consolidation problem is solved by forecasting CPU utilization on the basis of the Gray–Markov model. The objective of our approach is to minimize unnecessary VM migrations and the number of active hosts to economize on energy. Through our resource utilization prediction approach, this paper proposes a consolidation approach with UP-POD and UP-PUD algorithms for energy-efficient cloud data centers. With proper consideration of several aspects, the proposed approach effectively reduces the number of migrations, energy consumption of the hosts, and SLA violations. The results for different PlanetLab workload days verify that our approach reduces the energy consumption by switching idle hosts into low-power mode with an appropriate balance with the SLA.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Ashraf, Cost-Efficient Virtual Machine Management: Provisioning, Admission Control, and Consolidation (Ph.D. dissertation), Turku Centre for Computer Science (TUCS) Dissertations Number 183, Âbo, Finland, 2014.

[2] A. Ashraf, M. Hartikainen, U. Hassan, K. Heljanko, J. Lilius, T. Mikkonen, I. Porres, M. Syeed, S. Tarkoma, Introduction to cloud computing technologies, in: I. Porres, T. Mikkonen, A. Ashraf (Eds.), Developing Cloud Software: Algorithms, Applications, and Tools, Turku Centre for Computer Science (TUCS) General Publication Number 60, Âbo, Finland, 2013, pp. 1–41.

[3] P. Barham, B. Dragovic, K. Fraser, S. H, T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfield, Xen and the art of virtualization, in: Ninteenth acm Symposium on Operating Systems Principles, SOSP, 2003, pp. 164–177.

[4] A. Beloglazov, R. Buyya, Managing overload hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints, IEEE Trans. Parallel Distrib. Syst. (2012).

[5] A. Beloglazov, R. Buyya, Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers, Concurr. Comput.: Pract. Exper. (2012) 1397–1420.

[6] A. Beloglazov, R. Buyya, Y.C. Lee, A. Zomaya, A taxonomy and survey of energy-efficient data centers and cloud computing systems, Adv. Comput. 82 (2011) 47–111.

[7] R. Calheiros, R. Ranjan, A. Beloglazov, C.A.F. De Rose, R. Buyya, Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Softw. - Pract. Exp. 41 (2011) 23–50.

[8] C. Clark, K. Fraser, S. Hand, J.G. Hansen, E. Jul, C. Limpach, I. Pratt, A. Warfield, Live migration of virtual machines, in: Proc. 2nd Conf. Symp. Netw. Syst. Design Implementation, Vol. 2, 2005, pp. 273–286.

[9] L. Deboosere, B. Vankeirsbilck, P. Simoens, F. Turck, B. Dhoedt, P. Demeester, Efficient resource management for virtual desktop cloud computing, J. Supercomput. 62 (2) (2012) 741–767.

[10] J.L. Deng, Introduction to grey system theory, J. Grey Syst. 1 (1) (1989) 1–24.

[11] F. Farahnakian, P. Liljeberg, J. Plosila, Lircup: Linear regression based cpu usage prediction algorithm for live migration of virtual machines in data centers, in: The 39th Euromicro Conference Series on Software Engineering and Advanced Applications, 2013.

[12] T.C. Ferreto, M.A.S. Netto, R.N. Calheiros, C.A.F. De Rose, Server consolidation with migration control for virtualized data centers, Future Gener. Comput. Syst. 27 (2011) 1027–1034.

[13] X. Fu, C. Zhou, Virtual Machine selection and placement for Dynamic consolidation in cloud computing environment, Front. Comput. Sci. 9 (2) (2015) 322–330.

[14] A. Gandhi, M. Harchol-Balter, R. Das, et al., Optimal power allocation in server farms, in: Proceedings of the 11th International Joint Conference on Measurement and Modeling of Computer Systems, ACM New York, NY, USA, 2009, pp. 157–168.

[15] A. Greenberg, J. Hamilton, D.A. Maltz, P. Patel, The cost of a cloud: research problems in data center networks, ACM SIGCOMM Comput. Commun. Rev. 39 (2009) 63–73.

[16] G. Han, W. Que, G. Jia, L. Shu, An efficient virtual machine consolidation scheme for multimedia cloud computing, Sensors 16 (2) (2016) Article 246.

[17] Y. He, Y.D. Bao, A Grey–Markov forecasting model and its application, Syst. Eng. Theory Pract. 9 (4) (1992) 59–63.

[18] N. Hieu, M. Francesco, A. Yla-Jaaski, Virtual machine consolidation with usage prediction for energy-efficient cloud data centers, in: Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on, IEEE, 2015.

[19] J. Jheng, F. Tseng, H. Chao, Li-Der Chou, A novel VM workload prediction using grey forecasting model in cloud data center. in: Intl. Conference on Information Networking, 2014, pp. 40–45.

[20] U. Kumar, V.K. Jain, Time series models (Grey–Markov, Grey Model with rolling mechanism and singular spectrum analysis) to forecast energy consumption in India, Energy 35 (2010) 1709–1716.

[21] W. Li, J. Tordsson, E. Elmroth, Modeling for dynamic cloud scheduling via migration of virtual machines, in: Third IEEE International Conference on Cloud Computing Technology and Science, 2011.

[22] Z.-L. Mao, J.-H. Sun, Application of Grey–Markov model in forecasting fire accidents, Procedia Eng. 11 (2011) 314–318.

[23] C. Mastroianni, M. Meo, G. Papuzzo, Probabilistic consolidation of virtual machines in self-organizing cloud data centers, IEEE Trans. Cloud Comput. 1 (2013) 125–228.

[24] P. Mell, T. Grance, The NIST definition of cloud computing Recommendations of the National Institute of Standards and Technology. Special Publication 800-145 [Online], 2011, Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[25] G. Motta, N. Sfondrini, D. Sacco, Cloud computing: An architectural and technological overview, in: Proc. Int. Joint Conf. Serv. Sci. 2012, pp. 23–27.

[26] A. Murtazaev, S. Oh, Sercon: Server consolidation algorithm using live migration of virtual machines for green computing, J. ETE Tech. Rev. 28 (2011) 212–231.

[27] R. Nathuji, K. Schwan, Virtualpower: coordinated power management in virtualized enterprise systems, Oper. Syst. Rev. 41 (6) (2007) 265–278.

[28] K. Park, V.S. Pai, Comon: A mostly-scalable monitoring system for planetlab, ACM SIGOPS Oper. Syst. Rev. 40 (2006) 65–74.

[29] I. Sriram, A. Khajeh-Hosseini, Research agenda in cloud technologies, in: Large Scale Complex IT Syst., LSCITS, Univ. Bristol, U.K, 2010, http://arxiv.org/ftp/arxiv/papers/1001/1001.3257.pdf.

[30] I. Takouna, E. Alzaghoul, C. Meinel, Robust virtual machine consolidation for efficient energy and performance in virtualized data centers, in: The IEEE International Conference on Green Computing and Communications, 2014.

[31] X. Wang, M. Meng, Forecasting electricity demand using Grey–Markov Model, in: Proc. Seventh Int'l Conf. Machine Learning and Cybernetics, 2008, pp. 1244–1248.

[32] T. Wood, P. Shenoy, A. Venkataramani, M. Yousif, Black-box and gray-box strategies for virtual machine migration, in: The 4th USENIX conference on Networked systems design and implementation, 2007, pp. 229–242.

[33] Z. Xiao, W. Song, Q. Chen, Dynamic resource allocation using virtual machines for cloud computing environment, IEEE Trans. Parallel Distrib. Syst. 24 (2013) 1107–1116.

[34] X. Zhu, D. Young, B.J. Watson, Z. Wang, J. Rolia, S. Singhal, B. McKee, C. Hyser, et al., 1000 Islands: Integrated capacity and workload management for the next generation data center, in: Proceedings of the 5th Intl. Conf. on Autonomic Computing, ICAC, 2008, pp. 172–181.

Further reading

[1] S. Islam, J. Keung, K. Lee, A. Liu, Empirical prediction models for adaptive resource provisioning in the cloud, Future Gener. Comput. Syst. 28 (2012) 155–162.

Sun-Yuan Hsieh received the Ph.D. degree in computer science from National Taiwan University, Taipei, Taiwan, in June 1998. He then served the compulsory two-year military service. From August 2000 to January 2002, he was an assistant professor at the Department of Computer Science and Information Engineering, National Chi Nan University. In February 2002, he joined the Department of Computer Science and Information Engineering, National Cheng Kung University, and now he is a distinguished professor. He received the 2007 K. T. Lee Research Award, President's Citation Award (American Biographical Institute) in 2007, the Engineering Professor Award of Chinese Institute of Engineers (Kaohsiung Branch) in 2008, the National Science Council's Outstanding Research Award in 2009, and IEEE Outstanding Technical Achievement Award (IEEE Tainan Section) in 2011. He is Fellow of the British Computer Society (BCS). His current research interests include design and analysis of algorithms, fault-tolerant computing, bioinformatics, parallel and distributed computing, and algorithmic graph theory.

Cheng-Sheng Liu received the B.S. degree from the Department of Healthcare Administration and Medical Informatics, Kaohsiung Medical University, Taiwan, in 2014. He is now a Master student in the Institute of Medical Informatics, National Cheng Kung University, Taiwan.

Dr. Rajkumar Buyya is a Fellow of IEEE, Professor of Computer Science and Software Engineering, Future Fellow of the Australian Research Council, and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft, a spin-off company of the University, commercializing its innovations in Cloud Computing. He has authored over 500 publications and four text books including "Mastering Cloud Computing" published by McGraw Hill, China Machine Press, and Morgan Kaufmann for Indian, Chinese and international markets respectively. He also edited

several books including "Cloud Computing: Principles and Paradigms" (Wiley Press, USA, Feb 2011). He is one of the highly cited authors in computer science and software engineering worldwide (h-index=108, g-index=225, 55800+ citations). Microsoft Academic Search Index ranked Dr. Buyya as the world's top author in distributed and parallel computing between 2007 and 2015. "A Scientometric Analysis of Cloud Computing Literature" by German scientists ranked Dr. Buyya as the World's Top-Cited Author and the World's Most-Productive Author in Cloud Computing.

Software technologies for Grid and Cloud computing developed under Dr. Buyya's leadership have gained rapid acceptance and are in use at several academic institutions and commercial enterprises in 40 countries around the world. Dr. Buyya has led the establishment and development of key community activities, including serving as foundation Chair of the IEEE Technical Committee on Scalable Computing and five IEEE ACM conferences. These contributions and international research leadership of Dr. Buyya are recognized through the award of "2009 IEEE TCSC Medal for Excellence in Scalable Computing" from the IEEE Computer Society TCSC. Manjrasoft's Aneka Cloud technology developed under his leadership has received "2010 Frost and Sullivan New Product Innovation Award" and recently Manjrasoft has been recognised as one of the Top 20 Cloud Computing companies by the Silicon Review Magazine. He served as the foundation Editor-in-Chief of IEEE Transactions on Cloud Computing. He is currently serving as Co-Editor-in-Chief of Journal of Software: Practice and Experience, which was established 40+ years ago. For further information on Dr.Buyya, please visit his cyberhome: www.buyya.com.

Dr. Albert Y. Zomaya is the Chair Professor of High Performance Computing & Networking and Australian Research Council Professorial Fellow in the School of Information Technologies, Sydney University. He is also the Director of the Centre for Distributed and High Performance Computing which was established in late 2009. Dr. Zomaya published more than 500 scientific papers and articles and is author, co-author or editor of more than 20 books. He served as the Editor in Chief of the IEEE Transactions on Computers (2011–2014) and was elected recently as a Founding Editor in Chief for the newly established IEEE Transactions on Sustainable Computing.

Dr. Zomaya is the recipient of the IEEE Technical Committee on Parallel Processing Outstanding Service Award (2011), the IEEE Technical Committee on Scalable Computing Medal for Excellence in Scalable Computing (2011), and the IEEE Computer Society Technical Achievement Award (2014). He is a Chartered Engineer, a Fellow of AAAS, IEEE, and IET. Dr. Zomaya's research interests are in the areas of parallel and distributed computing and complex systems.